A STUDY ON RSA ALGORITHM

[1]Mayank Bhardwaj, [1]Manmohan Singh Rawat, [2]Sameer Dev Sharma

1. MCA Student - Uttaranchal Institute Of Management, Uttaranchal University, Arcadia Grant, Dehradun
2. Assistant Professor – Computer Applications, Uttaranchal University, Dehradun

Abstract

In world of dilation, where technology is on a run of tremendous growth and cybernated technologies are expanding rapidly, the cyberspace or commonly known as the world wide web has played a vital role in the distribution of digital archives such as audio files, portable documents, television broadcasts, graphical files and many more. A number of system have paved their way in making sure that the secure transmission of data takes place. The major limelight of the current research summary is on the study of the public key 'e' in the Rivest Shamir Adleman algorithm. The study is focused on the generation of several different values of public key and determine the result based on the value and speed of execution. The results depicts that modified algorithm provides additional security.

INTRODUCTION

The emergence of internet has been at a rapid speed but is still not 100% secured for interchange of data. To acquire user's trust , high speed complex algorithms have been developed around the world to enhance security[1].

Having a cryptosystem comprising of public key is preferred widely for safe data interchange as it comprises of encryption process, decryption process and authentication as well. Encryption basically deals with two types, i.e., symmetric and asymmetric encryption. Some examples of symmetric encryption are Blowfish, AES-128[2], AES-192[2], AES-256[2], DES, RC4, RC5, RC6 etc. Some examples of asymmetric encryption are DSA, Elliptic Curve technique, RSA[3], Diffie Helman[5], ECC[4] etc.

The RSA algorithm is the most suitable and preferred public key cryptography system because it provides better security by generating huge prime numbers for the modulus, private key and public key. The RSA algorithm is considered to be one of the most stable public key cryptosystem due to its retardation against conglomerate attacks[6].

However there are some drawbacks of RSA algorithm too. The ever growing demand of security by the users is one of the major challenges. The key areas where the RSA algorithm experiences hassle is it's slow speed and incompatibility to append with several systems. Keeping these drawbacks in mind, developers and researchers have proposed modifications which promises better security during transmission[1].

Multiple researches have been concluded with the modification of the modulus, i.e., 'n' in RSA algorithm[7][8][9][10]. However, researchers have neglected any manipulation regarding the public key in the RSA algorithm. Our review focuses on the alteration of the public key, i.e., 'e' and aims on illustrating its importance in the encryption and decryption technique.

RELATED LITERATURE

In order to modify the RSA algorithm, a number of studies and researches have been conducted. Some of those studies are mentioned below.

The modulus 'n' has been preferred to be replaced by a fresh element known as 'f'. This new element 'f' improvises the security regarding the factorization in the algorithm but at the same time also increases the consumption of time by the algorithm. A new study implemented a Binary cryptographic algorithm in modification to the RSA algorithm which improvised the security against the brute force attacks[13]. In this study, the algorithm would manipulate the message by the user into binary codes which proved to be more secure during the decryption phase. Although with increased security, this algorithm technique too consumed both time and space. Another study shows that a modified RSA algorithm made use of more than three prime numbers for the derivation of modulus 'n'[9][11]. This study proved to be helpful as the algorithm helped in repelling the factorization attacks. Another study showed a modified RSA algorithm which made use of two public keys at the same time[12]. These public keys improved the security to a great extent but also decreased the speed of algorithm massively.

APPROACH

Our study is focused on the comparison between the original and the manipulated RSA algorithm. The major idea behind this study is to understand the importance of public key, i.e., 'e' in the RSA algorithm.

The figure 1 shows the original process involved in the RSA algorithm.

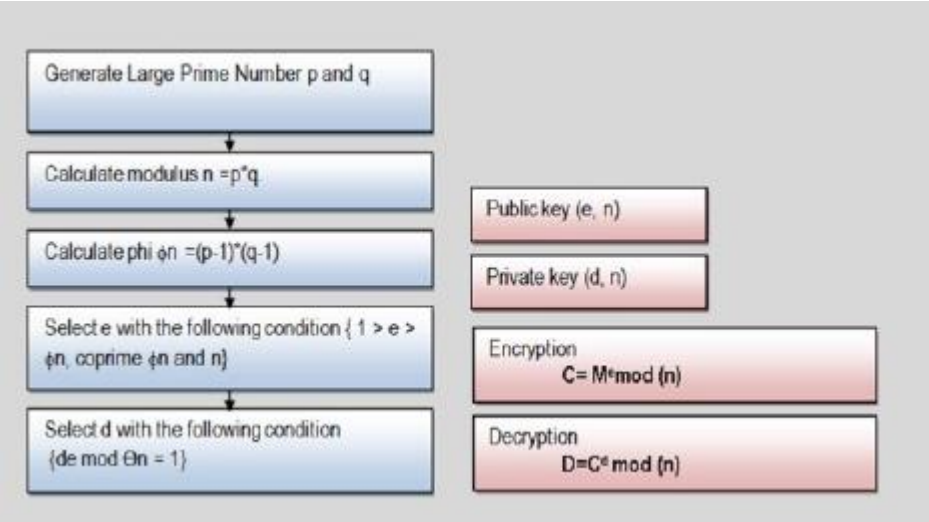


Figure 1

The figure 2 shows the process involved in the modified RSA algorithm.

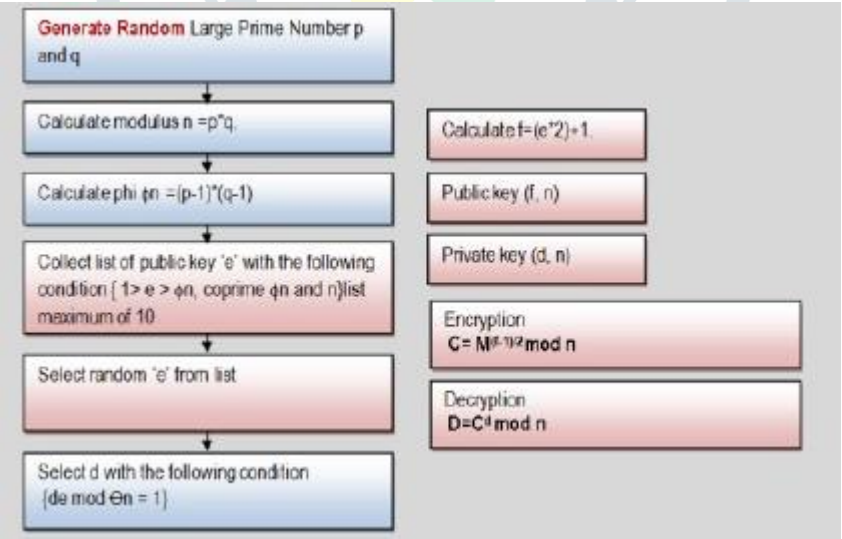


Figure 2

The figure 3 describes the modification of public key, i.e., 'e'.

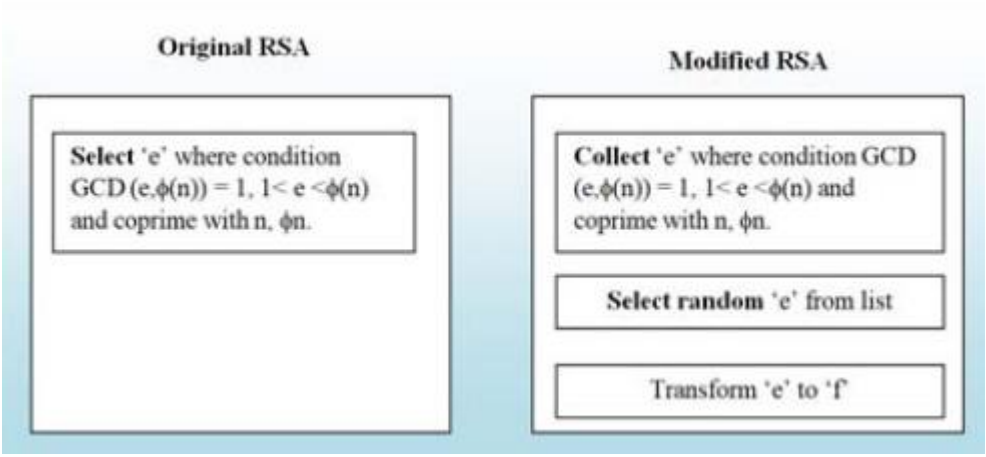


Figure 3[14]

The RSA algorithm consists of three processes, i.e., key generation, encryption and decryption.

Original RSA algorithm

Key Generation

1. Select two prime numbers 'p' and 'q'.
2. Calculate the value of n, i.e., n=p*q.
3. Calculate the value of ϕ(n), i.e., ϕ(n)=(p-1)*(q-1).
4. Choose a public key 'e'. The public key 'e' and the ϕ(n) must have common factor as 1 only. Also, 1<e<ϕ(n).
5. Calculate 'd' considering the condition (de mod ϕ(n))=1.

Encryption

1. Calculate M^e mod n.

Decryption

1. Calculate C^d mod n.

Modified RSA algorithm

Key Generation

1. Select two prime numbers 'p' and 'q'.
2. Calculate the value of n, i.e., n=p*q.
3. Calculate the value of ϕ(n), i.e., ϕ(n)=(p-1)*(q-1).
4. Compute ten value of 'e' considering the following condition p>e>ϕ(n) and ϕ(n) and n are the co-primes.
5. Select any random value of 'e' from the generated list.
6. Compute the value of f, i.e., f=(e*2)+1.
7. Calculate 'd' considering the condition (de mod ϕ(n))=1.
8. Dispatch public key (f,n).
9. Dispatch private key (d,n).

Encryption

1. Calculate $C=M^{((f-1)/2)}$ mod n.

Decryption

1. Calculate C^d mod n.

Example

Key Generation

1. We have considered two prime numbers 7 and 11 which are the values of p and q respectively.
2. The next step is to calculate the modulus. So, n = 7 X 11
   , i.e., n = 77.
3. Now find the value of ϕ(n), i.e., [7-1] X [11-1]
   So, ϕ(n) = 60.
4. We will generate 10 value of public key 'e'. Example : e=[13, 23, 27, 31, 37, ……, 61].
5. Now form the generated keys, select any value at random. Let us say 23.
6. Now calculate the value of 'f'. So, f = [23 X 2] – 1, i.e., the value of f is 47.
7. Now we'll calculate d considering the following condition.
   de mod n = 1.
   Therefore, the value of 'd' will be 47.
8. Our public key [f, n] = [47, 77].
9. Our private key [d, n] = [47, 77].

Encryption

From the formula mentioned above in the section of modified RSA algorithm, we have the following

M = 4

$C = [4^{(f-1)/2}]$ mod n

$C = [4^{23}]$ mod 77 => 9

Decryption

From the formula mentioned above in the section of modified RSA algorithm, we have the following

$D = C^d$ mod n

$D = [9^{47}]$ mod 77 => 4

RESULT

The original and the modified RSA algorithm, both were coded then in the JAVA programming language. The result varied for both the algorithms and efficiency was tested for both the algorithms.

The following table shows the various value of cipher texts whilst considering the public key 'e'.

p=5 q=11

e	d	cipher		
		m=4	m=5	m=6
13	37	9	15	51
17	33	49	25	41
23	7	9	15	51
27	3	49	25	41
29	29	14	25	29

p=11 q=17

e	d	cipher		
		m=4	m=5	m=6
3	107	64	125	29
7	23	115	146	184
13	37	174	37	95
19	59	47	108	46
29	149	157	146	104

The next two tables describes the different values of cipher text based on various values of 'e' between original and the modified RSA algorithm.

Result for p=5 q=11

e(Original RSA)	e(Modified RSA)	d	cipher	
			Original RSA	Modified RSA
13	27	37	9	49
17	35	33	49	0
23	47	7	9	0
27	55	3	49	0
29	59	29	14	0

Result for p=11 q=17

e(Original RSA)	e(Modified RSA)	d	Cipher	
			Original RSA	Modified RSA
3	7	107	64	115
7	15	23	115	166
13	27	37	174	115
19	39	59	47	0
29	59	149	157	0

The following graphs describes the difference in encryption time and decryption time between the original and the modified RSA algorithm.

Figure 1. Encryption Time

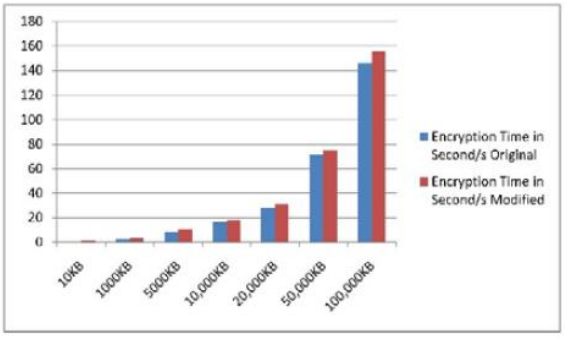
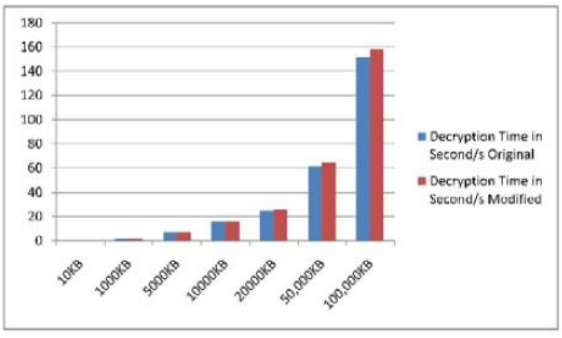


Figure 2. Decryption Time



The following table tells us about the difference in execution speed of the original and the modified RSA algorithm.

Result of Time Execution based on 'p' and 'q'

No	p	q	n	Time(second)			
				Encryption		Decryption	
				Original	Modified	Original	Modified
1	97	367	35,599	2	3	2	2
2	127	383	48,641	2	3	2	2
3	233	409	95,297	2	3	2	2
4	353	431	152,143	2	4	2	2
5	373	487	181,651	2	4	2	2
6	401	523	209,723	2	5	2	2
7	431	587	252,997	2	6	2	2
8	457	619	282,883	2	6	2	2
9	547	677	370,319	3	6	2	2
10	601	733	440,533	3	6	2	2

SUMMARY

		Original RSA Algorithm	Modified RSA Algorithm
Performance	encryption	√	
	decryption	√	√
Security	Public key		√
	Encrypted Message		√

The above table describes the comparison between the original and the modified RSA algorithm. It is clear that even at the cost of a little complexity, the modified RSA algorithm has proven out to be more secure considering the public key 'e' and the cipher message.

CONCLUSION

The conclusion of our study is basically based on the two important factors. They are the performance and the security provided by the algorithm. Between the original RSA algorithm and the modified RSA algorithm, we have inked our response as follows.

From the study concluded on the basis of performance, we've reached to this conclusion that the modified RSA algorithm has somewhere taken a blow in terms of time stamping but on the other hand also increased the security at the same time during the encryption process. Also, during the decryption phase, the modified RSA algorithm has shown a negligible increase or decrease in time as compared to the original RSA algorithm.

From the study concluded on the basis of security, the manipulated RSA algorithm gave out some cozy and complex results during the encryption process. Also, the use of various public keys, i.e., 'e' produced different values of cipher text which made sure that encrypted message was secure enough against attacks like brute force and factorization attacks during the decryption process.

At last considering all the results above generated in our study, we can say that, yes, the modification of public key 'e' plays an important role to enhance the security of the encrypting and the decrypting process in the modified RSA algorithm.

REFERENCES

[1]. D. I. G. Amalarethinam and J. S. Geetha, "A Survey on Secured Communication with High Speed using Public Key Cryptography".

[2]. N. Khanezaei and Z. M. Hanapi, "A framework based on RSA and AES encryption algorithms for cloud computing services," Proc. - 2014 IEEE Conf. Syst. Process Control. ICSPC 2014, no. December, pp. 58–62, 2014.

[3]. M. Wagner, and R. Bassous, "Multi -Asymmetric Cryptographic RSA Scheme," pp. 1–8, 2017.

[4]. P. Patel, R. Patel, and N. Patel, "Integrated ECC and Blowfish for Smartphone Security," in Physics Procedia, 2016.

[5]. W. Diffie, W. Diffie, and M. E. Hellman, "New Directions in Cryptography," IEEE Trans. Inf. Theory, 1976.

[6]. R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," Commun. ACM, 1978.

[7]. R. S. Dhakar, "Modified RSA Encryption Algorithm (MREA)," pp. 2–5, 2012.

[8]. J. Sahu, V. Singh, V. Sahu, and A. Chopra, "An Enhanced Version of RSA to Increase the Security," vol. 7, no. 4, pp. 1–4, 2017.

[9]. B. P. U. Ivy, P. Mandiwa, and M. Kumar, "A modified RSA cryptosystem based on ' n ' prime numbers," Int. J. Eng. Comput. Sci., 2012.

[10]. N. Kumar, "Implementation of Modified RSA Cryptosystem for Data Encryption and Decryption based on n Prime number and Bit Stuffing," 2016.
[11]. A. Shamir and L. Adleman, "Introduction," www.ijecs.in Int. J. Eng. Comput. Sci., vol. ISSN, no. 2, pp. 2319–7242, 2012.

[12]. A. A. Ayele, "A Modified RSA Encryption Technique Based on Multiple public keys," vol. 1, no. 4, pp. 859–864, 2013.

[13]. S. A. Nagar and S. Alshamma, "High Speed Implementation of RSA Algorithm with Modified Keys Exchange," pp. 639–642, 2012.

[14]. "Design and Implementation of RSA Algorithm using FPGA Council for Innovative Research," no. September, 2015.